REMARKS

This Amendment is filed in response to the Office Action dated May 30, 2008. All objections and rejections are respectfully traversed.

.

Claims 1-11, 13-26, and 28-49 are currently pending.

Claims 1, 11, 20, 21, 22, 32, 38, and 48 have been amended to better claim the invention.

No Claims have been added.

No Claims have been cancelled.

Request for Interview

The Applicant respectfully requests a telephonic interview with the Examiner after the Examiner has had an opportunity to consider this Amendment, but before the issuance of the next Office Action. The Applicant may be reached at 617-951-2500.

At Page 3, Paragraph 4 of the Office Action the claims were objected to. Amendment of the claims is believed to satisfy the objections.

At Paragraph 5 of the Office Action claims 11, 13-19, and 49 were rejected under 35 U.S.C. 103(a) as being unpatentable over Hitz et al. U. S. Patent 5,819,292 (hereinafter Hitz), in view of Bush et al. U. S. Patent 5,790,778 (hereinafter Bush).

Applicant's claimed invention, as set forth in representative claim 1, comprises in part:

1. A method for detecting leaked buffer writes between a first consistency point and a second consistency point in a data storage system, the method comprising:

receiving a write operation, the write operation identifying a file for the write operation;
determining that a volume storing the file has buffer leakage detection activated;

creating a data buffer associated with the write operation; and

in response to determining the volume has buffer leakage detection activated, writing a buffer check control structure to a raw data buffer associated with the data buffer, *the buffer check control structure including one or more uniquely identifying numbers referred to as magic numbers and a consistency point number, the magic numbers to uniquely identify the raw data buffer as a labeled buffer check control structure and to indicate that the data buffer needs to be checked for leakage*.

Hitz discloses a method for maintaining consistent states of a file system. In particular, incore WAFL Inodes contain information stemming from an on-disk inode, a WAFL buffer data structure, and buffer pointers. Specifically, this incore inode contains incore information pertaining to the buffer structure, where the incore information is made up of a dirty flag, an in-consistency point (IN_CP) flag and pointers for a linked list. The dirty flag represents an inode that has been modified or references buffers that have changed. The IN_CP flag, on the other hand, is used to mark an inode as being in a consistency point. Therefore, these flags are used to indicate when the inode is in a consistency point and must be cleared and written to a disk during its consistency point.

Bush discloses a first computer program which checks a second computer program for errors. The second computer program is executed by the first computer program, and various errors which occur during execution of the second computer program are recorded. The second computer program is executed over and over again for the purpose of detecting errors in operation of the second computer program. The first computer program uses a group of flags to indicate that an error has occurred in memory, at a location in memory referred to as a "chunk" of memory. Some of the flags are used

to determine if a chunk of memory has "leaked", as Bush describes at his Column 28, line 50-Column 29 line 15, which state:

Action 1504 creates a model for one or more contiguous memory locations. A memory location is the smallest unit of memory that can be explicitly and uniquely specified by means of an address. Typically, computer memory is byte addressable, and thus, a location is one byte. Action 1504 models memory using a chunk 1700. Chunk 1700 is shown in FIG. 17. Chunk 1700 includes fields: "freed flag" 1702, "reachable flag" 1704, "lost flag" 1706, "memory type" 1708, "chunk number" 1710, "origin context structure pointer" 1712, "stored value pointer" 1714 and "original stored value pointer" 1716.

Flag "freed flag" is true when the memory locations modeled by chunk 1700 have been freed. Flag "reachable flag" 1702 is used by leak detection processing to determine if the memory location is reachable. Flag "lost flag" 1706 is true when it can not be determined if the memory modeled is freed or **leaked**. With lost memory, it is possible that nothing will point to the memory after the function exits, but just because there is no record of a pointer to the memory does not mean that such a pointer does not exist. For example, memory can be allocated and then passed to a routine which is modeled by the missing model. Analysis engine 308 can not ascertain what happened to the allocated memory passed into the routine. Thus, the memory is marked as "lost". Field "memory type" 1708 holds the same information as field "memory type" 1606 described above. Field "chunk number" 1710 is a unique identifier for chunk 1700. Field "origin context structure pointer" 1712 points to the origin context structure 1600 created in action 1502. Field "stored value pointer" 1714 points to the current value in the modeled memory location. Field "original stored value pointer" 1716 points to the original value in the modeled memory location.

Bush gives as his definition of "leaked" memory his Col. 26 Lines 43-45, as:

A piece of memory is leaked when it is allocated, but it will not be pointed to by any symbol after the function exits.

Applicant respectfully notes that the Examiner found, at Page 4 last paragraph, through Page 5 , first 3 lines:

"Though Hitz discloses storing consistency point numbers in the buffer check control structure, he fails to further teach determining if one or more uniquely identifying numbers (hereinafter magic numbers) are within the data buffer check control structure, wherein the magic numbers are used to uniquely identify the raw data buffer and to indicate that the data buffer needs to be checked for leakage as recited in the instant claim."

Bush similarly fails to cure the deficiency of Hitz. Applicant respectfully urges that Bush has no disclosure of Applicant's claimed use of *including one or more uniquely identifying numbers referred to as magic numbers . . . the magic numbers to uniquely identify the raw data buffer as a labeled buffer check control structure and to indicate that the data buffer needs to be checked for leakage.*

Applicant uses his *one or more uniquely identifying numbers referred to as magic numbers* during real-time operation of a *data storage system*, as is shown by Applicant's claimed *receiving a write operation, the write operation identifying a file for the write operation*;

 determining that a volume storing the file has buffer leakage detection activated;

 creating a data buffer associated with the write operation.

In sharp contrast, Bush has his first computer program analyzing his second computer program over and over again, for different sets of parameters, to determine if any errors occur..

Accordingly, Applicant respectfully urges that the use by Bush of his repetitive testing of his second computer program cannot, under 35 U.S.C. 103(a) render Applicant's claimed invention obvious, because Bush has no disclosure of Applicant's claimed real time *including one or more uniquely identifying numbers referred to as*

magic numbers . . . the magic numbers to uniquely identify the raw data buffer as a labeled buffer check control structure and to indicate that the data buffer needs to be checked for leakage.

That is, Applicant checks for memory leakage while a computer program is running. Bush simply executes a program over and over again, and checks for errors after the program has executed many times.

Further, the combination of Hitz and Bush teaches away from Applicant's claimed novel invention. Particularly, Bush teaches that to detect errors in operation of a computer program a person must execute the program over and over again by a controlling program. Hitz is simply silent concerning Applicant's claimed novel invention. Thus, a person of ordinary skill in the art of computer programming would be led astray from Applicant's claimed invention, by the person expecting to have to have a controlling program repeatedly executing a second program to find errors.

Such a person of ordinary skill in the art would be led astray from Applicant's claimed novel use of Applicant's *magic numbers . . . the magic numbers to uniquely identify the raw data buffer as a labeled buffer check control structure and to indicate that the data buffer needs to be checked for leakage* , which detect errors during real time operation of a computer program, as Applicant's claimed *receiving a write operation, the write operation identifying a file for the write operation*;

> *determining that a volume storing the file has buffer leakage detection activated*;

> *creating a data buffer associated with the write operation* sets out.

Accordingly, Applicant respectfully urges that neither Hitz nor Bush, either singly or in combination, are legally capable of rendering Applicant's claimed novel invention

unpatentable under 35 U.S.C. 103(a) because of the absence from both of Applicant's claimed *receiving a write operation*, the write operation identifying a file for the write operation;

determining that a volume storing the file has buffer leakage detection activated . . . in response to determining the volume has buffer leakage detection activated . . . the buffer check control structure including one or more uniquely identifying numbers referred to as magic numbers and a consistency point number, the magic numbers to uniquely identify the raw data buffer as a labeled buffer check control structure and to indicate that the data buffer needs to be checked for leakage.

At Paragraph 6 of the Office Action, claims 1-10, 20-26, and 28-48 were rejected under 35 U.S.C. 103(a) as being unpatentable over Hitz in view of Marion et al. U. S. Patent Publication Number 2003 / 0163661 (hereinafter Marion), and further in view of Bush.

Marion discloses a system for detecting memory leaks caused by allocating a region of memory and then failing to de-allocate that region. Marion sets a flag to indicate that the region was allocated, and then later checks to see if the flag is still set. If the flag is still set, Marion then de-allocates the memory region.

Applicant respectfully urges that Marion has no disclosure of Applicant's claimed *receiving a write operation*, the write operation identifying a file for the write operation;

determining that a volume storing the file has buffer leakage detection activated . . . in response to determining the volume has buffer leakage detection activated . . . the buffer check control structure including one or more uniquely identifying numbers referred to as magic numbers and a consistency point number, the magic numbers to uniquely identify the raw data buffer as a labeled buffer check control structure and to indicate that the data buffer needs to be checked for leakage.

That is, Applicant identifies, by using *including one or more uniquely identifying numbers* in order to *indicate that the data buffer needs to be checked for leakage*.

Further, Marion, in combination with Hitz and Bush teach away from Applicant's claimed invention.

A person of ordinary skill in the art would be led astray from Applicant's claimed novel invention by following the teachings of Marion, in that in following Marion the person would believe that he has to set flags and clear flags as memory is allocated and later is de-allocated, and then later look for flags which have not been cleared.

That is, Marion has no disclosure of Applicant's claimed novel *the buffer check control structure including one or more uniquely identifying numbers referred to as magic numbers and a consistency point number, the magic numbers to uniquely identify the raw data buffer as a labeled buffer check control structure and to indicate that the data buffer needs to be checked for leakage*.

Accordingly, Applicant respectfully urges that a combination of Hitz, Bush, and Marion is legally incapable of rendering Applicant's claimed invention obvious under 35 U.S.C. 103(a) because of the absence from each of Applicant's claimed *receiving a write operation*, the write operation identifying a file for the write operation;

determining that a volume storing the file has buffer leakage detection activated . . . in response to determining the volume has buffer leakage detection activated . . . the buffer check control structure including one or more uniquely identifying numbers referred to as magic numbers and a consistency point number, the magic

numbers to uniquely identify the raw data buffer as a labeled buffer check control structure and to indicate that the data buffer needs to be checked for leakage.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

  /A. Sidney Johnston/
A. Sidney Johnston
Reg. No. 29,548
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500